Amendments to the Claims

Please amend claims as follows.

1. (Previously presented) An apparatus for generating <u>processor-specific multimedia routines dynamically</u> ~~computer assembly code~~, comprising:

    <u>a computer; and</u>

    <u>a program executing on said computer, said program including instructions for</u>

5   <u>processing multimedia data, said program further including</u>:

    ~~and~~ an abstract routine generator for receiving a data stream comprising a multimedia routine and for outputting a non-processor-specific abstract representation thereof ~~during runtime~~ <u>at program startup</u>; and

    a translator for said abstract routine generator for receiving said abstract

10   representation and for outputting processor-specific final code translated from said non-processor-specific abstract representation for processing multimedia input data ~~during said runtime~~ <u>at program startup</u>.

2. (original) The apparatus of Claim 1, where in said abstract routine generator builds an abstract routine during runtime.

15 3. (original) The apparatus of Claim 1, wherein said abstract routine generator builds an abstract routine in the form of a graph.

4. (original) The apparatus of Claim 1 wherein said multimedia data comprise SIMD input data.

20

5. (original) The apparatus of Claim 1, wherein said multimedia data comprise image input data.

6. (original) The apparatus of Claim 1, wherein said multimedia data comprise audio

25   input data.

7. (original) The apparatus of Claim 3, wherein said graph is input to said translator.

2

8. (Original) The apparatus of Claim 3, wherein the output of said translator is in assembly code.

9. (Cancelled)

5

10. (original) The apparatus of Claim 1, wherein said processor-specific code performs any of the operations of add, sub, multiply, average, maximum, minimum, compare, and, or, xor, pack, unpack, and merge on said input data.

10   11. (original) The apparatus of Claim 3, wherein said graph is a function of any of source block, target block, change in the block, color, stride, change in stride, display block, and spatial filtering.

12. (Currently Amended) A method for generating <u>processor-specific multimedia</u>
15   <u>routines dynamically</u> ~~computer-assembly-code~~, comprising:

    providing <u>a computer; and</u>

        <u>a program executing on said computer, said program including</u>
        <u>instructions for processing multimedia data, said program further</u>
        <u>including</u>:

20   ~~providing~~ an abstract routine generator for receiving a data stream comprising a multimedia routine and for outputting a non-processor-specific abstract representation thereof ~~during runtime~~ <u>at program startup</u>; and

        ~~providing~~ a translator for said abstract routine generator<u>:</u>
25   ~~for~~ receiving said abstract representation<u>;</u>
        and ~~for~~

    outputting processor-specific final code translated from said non-processor-specific abstract representation for processing multimedia input data ~~during said runtime~~ <u>at program startup</u>.

3

13. (Original) The method of Claim 12, wherein said abstract routine generator builds the abstract routine during runtime.

5 14. (Original) The method of Claim 13, wherein said abstract routine is a graph.

15. (original) The method of Claim 12, wherein said multimedia input data comprise SIMD data.

10 16. (original) The method of Claim 12, said multimedia input data comprise image data.

17. (original) The method of Claim 12, wherein said multimedia Input data comprise audio data.

15 18. (original) The method of claim 14, wherein said graph is input to said translator.

19. (original) The method of claim 12, wherein the output of said translator is assembly code.

20 20. (original) The method of Claim 12, wherein said processor-specific code performs any of the operations of add, sub, multiply, average, maximum, minimum, compare, and, or, xor, pack, unpack, and merge on said multimedia input data.

21. (original) The method of Claim 14, wherein said graph is a function of any of source
25 block, target block, change in the block, color, stride, change in stride, display block, and spatial filtering.

22. (Cancelled)

4